

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Politehnica Timișoara
1.2 Facultatea ¹ / Departamentul ²	Inginerie Hunedoara/Inginerie Electrică și Informatică Industrială
1.3 Domeniul de studii (denumire/cod ³)	Inginerie electrică / 90
1.4 Ciclul de studii	Licență
1.5 Programul de studii (denumire/cod/calificarea)	Inginerie electrică și calculatoare / 60 / Inginer

2. Date despre disciplină

2.1a Denumirea disciplinei/Categoria formativă ⁴	Programare orientată pe obiecte / DS						
2.1b Denumirea disciplinei în limba engleză	Object Oriented Programming						
2.2 Titularul activităților de curs	Prof.dr.ing. Pănoiu Manuela						
2.3 Titularul activităților aplicative ⁵	Șef lucr. Dr. Ing. Ghiormez Loredana						
2.4 Anul de studii ⁶	II	2.5 Semestrul	2	2.6 Tipul de evaluare	E	2.7 Regimul disciplinei ⁷	DI

3. Timp total estimat - ore pe semestru: activități didactice directe (asistate integral sau asistate parțial) și activități de pregătire individuală (neasistate)⁸

3.1 Număr de ore asistate integral/săptămână	4 , format din:	3.2 ore curs	2	3.3 ore seminar/laborator/proiect	2
3.1* Număr total de ore asistate integral/sem.	56 , format din:	3.2* ore curs	28	3.3* ore seminar/laborator/proiect	28
3.4 Număr de ore asistate parțial/săptămână	, format din:	3.5 ore practică		3.6 ore elaborare proiect de diplomă	
3.4* Număr total de ore asistate parțial/semestru	, format din:	3.5* ore practică		3.6* ore elaborare proiect de diplomă	
3.7 Număr de ore activități neasistate/săptămână	3.14 , format din:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			1
		ore studiu individual după manual, suport de curs, bibliografie și notițe			1
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			1.14
3.7* Număr total de ore activități neasistate/semestru	44 , format din:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			14
		ore studiu individual după manual, suport de curs, bibliografie și notițe			14
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			16
3.8 Total ore/săptămână ⁹	7.14				
3.8* Total ore/semestru	100				
3.9 Număr de credite	4				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	• Programarea calculatoarelor și limbaje de programare 2
4.2 de rezultatele învățării	•

5. Condiții (acolo unde este cazul)

5.1 de desfășurare a cursului	• Sală de curs echipată cu videoproector și conexiune la Internet.
5.2 de desfășurare a activităților practice	• Sală de laborator echipată cu videoproector și computere.

6. Rezultatele învățării la formarea cărora contribuie disciplina

Cunoștințe	• C3. Studentul/absolventul explică și interpretează desenele care detaliază proiectarea produselor, a instrumentelor și a sistemelor de inginerie electrică
Abilități	• A3. Studentul/absolventul specifică proprietăți tehnice ale bunurilor, materialelor, metodelor, proceselor, serviciilor, sistemelor, software-ului și funcționalităților, prin identificarea și răspunsul la nevoile particulare care urmează să fie satisfăcute în funcție de cerințele clienților • A5. Studentul/absolventul desenează schițe și proiectează sisteme, produse și componente electrice utilizând programe și echipamente informatice de proiectare asistată de calculator (CAD). Desenează schițe ale panourilor electrice, scheme electrice, diagrame de cablare electrică și alte detalii ale ansamblului. Creează schițe și desene tehnice prin utilizarea de software specializat.
Responsabilitate și autonomie	• RA2. Studentul/absolventul efectuează căutări bibliografice în literatura de specialitate, consultă și folosește bazele de date științifice și alte surse de informare din domeniul ingineriei electrice. • RA5. Studentul/absolventul reflectă în mod critic, reflexiv, cu simțul responsabilității și în spirit democratic asupra responsabilităților etice și sociale legate de managementul activităților din domeniul ingineriei electrice, de luarea deciziilor și de formularea opiniilor •

7. Obiectivele disciplinei (asociate rezultatelor învățării de la punctul 6)

- Obiectivul acestei discipline îl reprezintă însușirea conceptelor specifice unui limbaj de programare orientat pe obiecte, cum ar fi: tip abstract de date, clasă, obiect, încapsulare, moștenire, derivare și polimorfism.
- Obiectivele specifice ale acestei discipline sunt:
- dezvoltarea deprinderilor de a rezolva probleme de dimensiuni mici și medii într-o manieră orientată pe obiecte;
 - dezvoltarea deprinderilor de a evidenția diferența între proiectarea procedurală și proiectarea orientată pe obiecte;
 - dezvoltarea deprinderilor de a să înțeleagă rolul moștenirii, polimorfismului, legării dinamice și a structurilor generice în dezvoltarea unor programe reutilizabile;
 - dezvoltarea deprinderilor de a implementa programe în limbajul de programare C++ de dimensiuni mici/medii;
- dezvoltarea deprinderilor de a utiliza clase scrise de alți programatori

8. Conținuturi¹⁰

8.1 Curs	Număr de ore	Metode de predare ¹¹
1. Introducere în programarea orientată pe obiect 1.1. Declararea și implementarea unei clase. Pointerul this 1.2. Protejarea informațiilor din cadrul obiectelor. Încapsulare 1.3. Funcții membre inline	2	Expunerea informațiilor esențiale pe videoprojector. Detalieră informațiilor expuse Demonstrarea utilizând un mediu de programare C/C++ Conversația
2. Crearea, instanțierea, inițializarea și distrugerea obiectelor 2.1. Liste de instanțieri 2.2. Constructori. Tipuri de constructori 2.3. Destructorii 2.4. Manevrarea dinamică a obiectelor	2	
3. Elemente caracteristice claselor 3.1. Date și funcții membre statice 3.2. Obiecte constante. Funcții membre constant 3.3. Funcții și clase prietene 3.4. Clase imbricate. Clase locale 3.5. Transmiterea obiectelor către funcții. Returnarea obiectelor de către funcții	4	
4. Moștenire și polimorfism 4.1. Derivarea claselor. Reutilizarea funcțiilor din clase derivate 4.2. Polimorfismul. Supraîncărcarea și supradefinirea funcțiilor.	6	

Funcții virtuale 4.3. Protecția informației într-o ierarhie de clase 4.4. Clase virtuale. Clase abstracte 4.5. Supraîncărcarea operatorilor		
5. Tratarea excepțiilor. Sistemul de intrare/ieșire din C++ 5.1. Declararea excepțiilor. Ierarhii de excepții 5.2. Ierarhia de clase corespunzătoare stream-urilor 5.3. Stream-uri predefinite în C++ 5.4. Redirecționări. Supraîncărcarea operatorilor << și >> 5.5. Operații de intrare/ieșire cu fișiere în C++	4	
6. Programare generică în C++. Biblioteca standard C++ 6.1. Funcții și clase generice în C++ 6.2. Biblioteca standard C++ 6.2.1. Containere 6.2.2. Iteratori 6.2.3. Algoritmi 6.2.4. Functori. Clase de functori	6	
7. Biblioteca Qt 7.1. Arhitectura Qt. Biblioteca de clase Qt 7.2. QtCore 7.3. QtGui	4	
Bibliografie ¹² 1. Manuela Pănoiu, Anca-Elena Iordan, Programare orientată pe obiecte în C++, Editura Politehnica, Timișoara, 2014 2. Liviu Negrescu, Limbajele C și C++ pentru începători, Volumul II – Limbajul C++, Editura Alabastră, Cluj-Napoca, 2006 3. Scott Meyers, STL – Biblioteca programatorului, Editura Teora, București, 2002 4. Constantin Gălățan, C++ - Introducere în Standard Template Library, Editura All, București, 2008		
8.2 Activități aplicative¹³	Număr de ore	Metode de predare
1. Analiza, proiectarea și implementarea de aplicații care utilizează tipuri de date abstracte definite prin intermediul structurilor.	4	Verificare cunoștințe din tematica laboratorului Se va utiliza exercițiul la tablă și implementarea programului utilizând calculatorul.
2. Analiza, proiectarea și implementarea de aplicații care utilizează tipuri de date abstracte definite prin intermediul claselor		
3. Analiza, proiectarea și implementarea de aplicații care utilizează constructori și destructori	2	
4. Analiza, proiectarea și implementarea de aplicații care utilizează date și funcții statice. Funcții prietene	2	
5. Analiza, proiectarea și implementarea de aplicații care utilizează compunerea de obiecte.	4	
6. Analiza, proiectarea și implementarea de aplicații care utilizează conceptul de moștenire.		
7. Analiza, proiectarea și implementarea de aplicații care utilizează funcții virtuale pure. Clase abstracte	2	Verificare cunoștințe din tematica laboratorului Se va utiliza exercițiul la tablă și implementarea programului utilizând calculatorul.
8. Analiza, proiectarea și implementarea de aplicații care utilizează supraîncărcarea operatorilor	4	Verificare cunoștințe din tematica laboratorului Se va utiliza exercițiul la tablă și implementarea programului utilizând calculatorul.
9. Analiza, proiectarea și implementarea de aplicații care utilizează ierarhii de clase pentru operații de intrare/ieșire	2	Verificare cunoștințe din tematica laboratorului Se va utiliza exercițiul

		la tablă și implementarea programului utilizând calculatorul.
10. Analiza, proiectarea și implementarea de aplicații prin reutilizarea codului sursă cu ajutorul claselor șablon	6	Verificare cunoștințe din tematica laboratorului Se va utiliza exercițiul la tablă și implementarea programului utilizând calculatorul.
11. Analiza, proiectarea și implementarea de aplicații prin reutilizarea codului sursă cu ajutorul containerelor secvențiale STL	2	Verificare cunoștințe din tematica laboratorului Se va utiliza exercițiul la tablă și implementarea programului utilizând calculatorul.
Bibliografie¹⁴ 1. Ghiormez L., Laborator în format electronic – Programare orientată pe obiecte, Campusul virtual al UPT, https://cv.upt.ro/course/view.php?id=2617 sau intranet facultate \\fs\cursuri\PROGRAMAREA CALC SI LIMBAJE DE PROGRAMARE - II IIND + II IEC 2025 2 Ghiormez L., Pănoiu M., Programarea calculatoarelor în limbajul C/C++, Ed. POLITEHNICA, Timișoara, 326 pg., 2022, ISBN: 978-606-35-0376-4 3. Iordan A., Cristea D., Berdie A., Programarea calculatoarelor, îndrumător de laborator, Editura Mirton, Timișoara, 2004 4. L. Negrescu, Limbajele C și C++ pentru începători, Volumul I – Limbajul C, Editura Albastră, Cluj-Napoca, 2009		

9. Evaluare

Tip activitate	9.1 Criterii de evaluare ¹⁵	9.2 Metode de evaluare	9.3 Pondere din nota finală
9.4 Curs	- Înțelegerea principiilor programării orientate pe obiecte și a conceptelor limbajului C++. - Cunoașterea sintaxei și a mecanismelor limbajului C++. - Capacitatea de proiectare logică a aplicațiilor orientate pe obiecte	Examen final (scris sau pe calculator) pentru verificarea cunoștințelor teoretice și aplicative	66%
9.5 Activități aplicative	S:		
	L: Abilități în analiza, proiectarea și implementarea aplicațiilor de laborator. Teme de casă și/sau în timpul desfășurării activității practice	Oral – aplicații utilizând calculatorul	34 %
	P¹⁶:		
	Pr:		
9.6 Standard minim de performanță (se prezintă cunoștințele minim necesare pentru promovarea disciplinei și modul în care se verifică stăpânirea lor ¹⁷)			
• Laborator: Rescrierea unei declarații de clasă în forma corectă sau obținerea a minim 50% din punctajul total alocat subiectelor • Curs: Realizarea unei aplicații C++ funcționale, care implementează clase și funcții proprii și demonstrează înțelegerea principiilor POO (încapsulare, moștenire, polimorfism))			

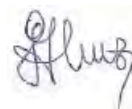
Data completării

10.09.2025

Titular de curs
(semnătura)



Titular activități aplicative
(semnătura)



Director de departament

Data avizării în Consiliul Facultății¹⁸

Decan

(semnătura)



17.09.2025

(semnătura)

